What is claimed is:

1. A method of sharing a memory module between a plurality of processors comprising:

dividing the memory module into n banks, where n = at least 2, wherein each bank can be accessed by one or more processors at any one time;

mapping the memory module to allocate sequential addresses to alternate banks of the memory; and

storing data bytes in memory, wherein said data bytes in sequential addresses are stored in alternate banks due to the mapping of the memory.

2. The method of claim 1 further including a step of dividing each bank into x blocks, where x = at least 1, wherein each block can be accessed by one of the plurality of processors at any one time.

3. The method of claim 1 or 2 further including a step of determining whether memory access conflict has occurred, wherein two or more processors are accessing the same block at any one time.

4. The method of claim 1, 2 or 3 further including a step of synchronizing the processors to access different blocks at any one time..

5. The method of claim 4 further including a step of determining access priorities of the processors when memory access conflict occurs.

6. The method of claim 5 wherein the step of determining access priorities comprises assigning lower access priorities to processors that have caused the memory conflict.

7. The method of claim 5 wherein the step of determining access priorities comprises assigning lower access priorities to processors that performed a jump.

8. The method of claim 4, 5, 6 or 7 wherein the step of synchronizing the processors comprises locking processors with lower priorities for one or more cycles when memory access conflict occurs.

9. A system comprising:

a plurality of processors;

a memory module comprising n banks, where n = at least 2, wherein each bank can be accessed by one or more processors at any one time;

a memory map for allocating sequential addresses to alternate banks of the memory module; and

data bytes stored in memory, wherein said data bytes in sequential addresses are stored in alternate banks according to the memory map.

10. The system of claim 9 wherein each bank comprises x blocks, where x = at least 1, wherein each block can be accessed by one of the plurality of processors at any one time.

11. The system of claim 9 or 10 further comprising a flow control unit for synchronizing the processors to access different blocks at any one time.

12. The system of claim 9, 10 or 11 further comprising a priority register for storing the access priority of each processor.

13. The system of any of claims 9-12 wherein said data bytes comprise program instructions.

14. The system of any of claims 10-13 further comprising a plurality of critical memory modules for storing a plurality of data bytes for each processor for reducing memory access conflicts.

15. A method of sharing a memory module between a plurality of processors comprising:

dividing the memory module into n banks, where n = at least 2, enabling the memory module to be accessed by one or more processors simultaneously;

mapping the memory module to allocate sequential addresses to alternate banks of the memory;

storing data words in memory, wherein data words in sequential addresses are stored in alternate banks due to the mapping of the memory; and

providing a first signal path, the first signal path coupling a cache to a processor and the memory module when selected, the cache enabling the processor to fetch a plurality of data words from different banks simultaneously.

16. The method of claim 15 further including a step of dividing the bank into x blocks, where x = at least 1, wherein a block can be accessed by one of the plurality of processors at any one time.

17. The method of claims 15 or 16 further including a step of determining whether contention has occurred, wherein two or more processors are accessing the same address range at any one time.

18. The method of claim 17 wherein the address range coincides with at least one block.

19. The method of any of the claims 15-18 further including a step of synchronizing the processors to access different banks when contention has occurred.

20. The method of any of the claims 15-19 further including the step of providing a second signal path, the second signal path coupling the processor to the memory module when selected.

21. The method of any of the claims 15-20 further including a step of activating the second signal path when contention has not occurred.

22. The method of any of the claims 15-21 further including a step of synchronizing the processors to access different banks when contention has occurred.

23. The method of any of the claims 15-22 further including a step of determining access priorities of the processors when contention has occurred.

24. The method of claim 23 wherein the step of determining access priorities comprises assigning lower access priorities to processors that have caused the contention.

25. The method of any of the claims 19-24 wherein the step of synchronizing the processors comprises inserting wait states for processors with lower priorities when contention occurs.

26. The method of any of the claims 15-25 further including a step of activating the first signal path when contention has occurred.

27. A system comprising:

a plurality of processors;

a memory module comprising n banks, where n = at least 2, wherein a bank can be accessed by one or more processors at any one time;

a memory map for allocating sequential addresses to alternate banks of the memory module;

data words stored in memory, wherein data words in sequential addresses are stored in alternate banks according to the memory map; and

a plurality of control logic unit for enabling a processor to access a plurality of data words from different banks.

28. The system of claim 27 wherein a control logic unit comprises first and second signal paths, the first signal path coupling a cache to a processor and the memory module, the second signal path coupling the processor to the memory module.

29. The system of claim 27 or 28 wherein the first signal path comprises a cache register and a multiplexer.

30. The system of any of the claims 27-29 wherein the bank comprises x blocks, where x = at least 1, wherein a block can be accessed by one of the plurality of processors at any one time.

31. The system of any of the claims 27-30 further comprising a flow control unit for synchronizing the processors to access different blocks at any one time.

32. The system of any of the claims 27-31 further comprising a priority register for storing the access priority of a processor.

33. The system of any of the claims 27-32 further comprising a plurality of critical memory modules for storing a plurality of data words for the processors to reduce the possibility of contention.

34. The system of any of the claims 27-33 wherein a control logic unit comprises a first signal path, the first signal path coupling a cache to a processor and the memory module.